Fig. 1
Example of STIL(1)

Signals:

```
Signals {
      Signal[0] InOut;
      Signal[1] InOut;
      Signal[2] InOut;
}

SignalGroups {
      signal = 'Signal[0..2]';
}
```

Fig. 2
Example of STIL(2)

Timing:

```
Timing T1
{
    WaveformTable wft1
    {
        Period '1us';
        Waveforms
        {
            Signal[0] {01 { '400ns' D/U; }}
            Signal[1] {01 { '400ns' D/U; '500ns' D;}
                23 { '0ns' U/D; '200ns' D/U; '600ns' U/D;}}
            Signal[2] {01  { '200ns' D/U; }
                23 {'100ns' D/U; '400ns' D; }}
        } // end of Waveforms block
    } // End of WaveformTable wft1
} // End of Timing block
```

Fig. 3
Example of STIL(3)

Pattern:

```
Pattern pat1 {

        W wft1;

        V { signal  =011; } // Note: signal = Signal[0] +
        V { signal  =020; } // Signal[1] + Signal[2]
        V { signal  =111; }
        V { signal  =103; }
        V { signal  =132; }
        V { signal  =000; }
        V { signal  =003; }
}
```

Fig. 4
Example of STIL(4)

Flow:

```
PatternBurst pb1 {
      PatList {
            pat1;
      } // end of PatList
} // end of PatternBurst

PatternExec {
      Timing T1;
      PatternBurst pb1;
} // end of PatternExec
```

Fig. 5
Example of TDL

Timing and Pattern:

```
SIGNAL signal_1;
signal_1.drekind( 0, NRZ);
signal_1.wavekind( 1, RZO);
signal_1.timing( 1, T1, 400.0nS);
signal_1.timing( 1, T2, 500.0nS);
signal_1.wavekind( 2, XOR);
signal_1.timing( 2, T3, 0.0nS);
signal_1.timing( 2, T1, 200.0nS);
signal_1.timing( 2, T2, 600.0nS);
signal_1.wavekind( 3, NRZ);
signal_1.wavekind( 4, NRZ);
signal_1.timing( 4, STBL, 0.0nS);
signal_1.timing( 4, DREL, 0.0nS);
```

Fig. 6A
Conversion Basics

- Vector-based to vector-based conversion

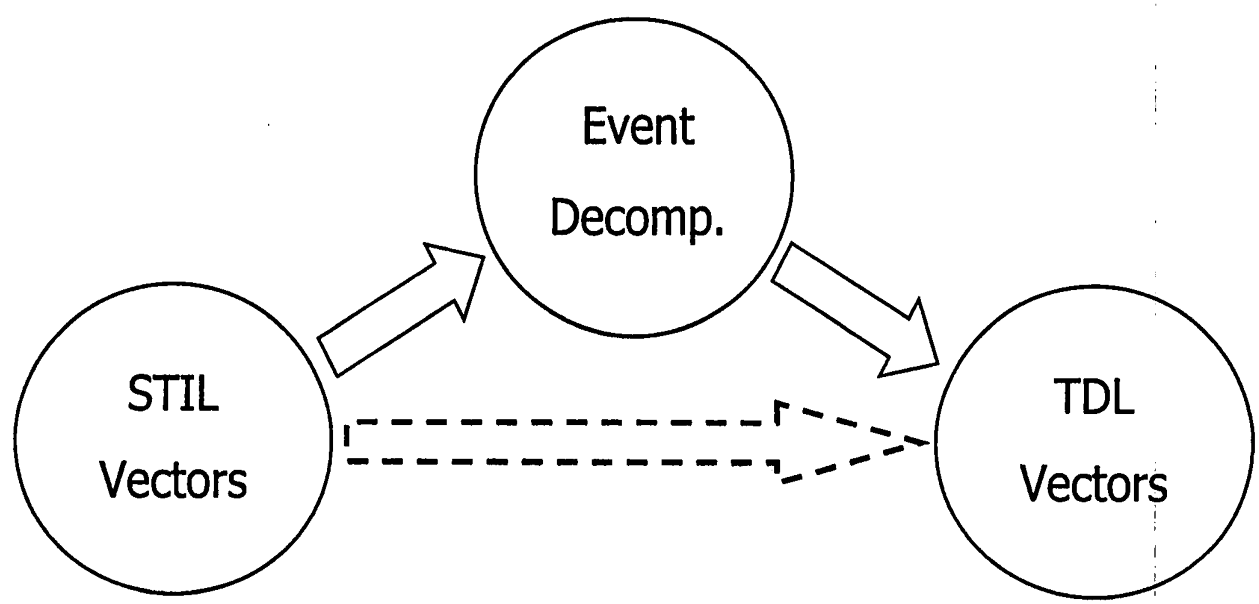


Fig. 6B

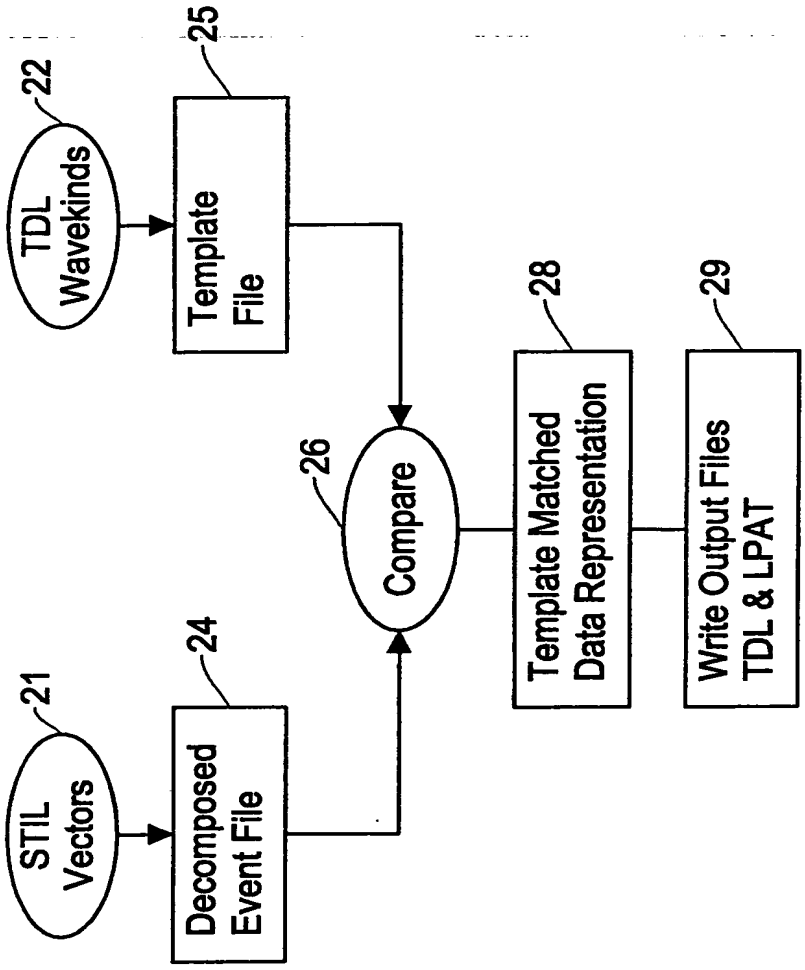

22
TDL Wavekinds
25
Template File
21
STIL Vectors
24
Decomposed Event File
26
Compare
28
Template Matched Data Representation
29
Write Output Files TDL & LPAT

Fig. 7
Template Matching

- 01 {'400ns' D/U;} => NRZ; T1 = 400ns; T2 = 400ns

- Template

```
{0,0,0}
{0,1,1 T2}
{1,0,1 T1}
{1,1,0}
```

Fig. 8
Wavekind Matching (1)

01 {'400ns' D/U;}

NRZ; T1=T2=400ns

0 {} 1{'200ns' U;
'400ns' D;}

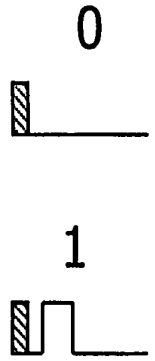
RZ0; T1=200ns;
T2=400ns

01{'100ns' D;
'200ns' D/U;
'400ns' D;}

?

PCT/US00/26189

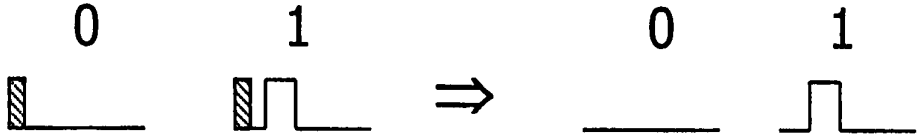
Fig. 9
Wavekind Matching (2)



NRZ; RZO...

XOR; T3=100ns; T1=200ns; T2=400ns

- Analyze timing block
  - What if no waveform can end in 'U' state?



RZ0; T1=200ns; T2=400ns

PCT/US00/26189

Fig. 10A

Number of Edges	Start Value 0	Start Value 1
0	T	T
1	F	T
2	T	F
3	F	F
4	F	F

0{}1{'200ns' U:'400ns'D;}

Fig. 10B

Number of Edges	Start Value 0	Start Value 1
0	T	T
1	F	T
2	T	F
3	F	T
4	F	F

23{'100ns' D; '200ns' D/U; '400ns' D;}

Fig. 11

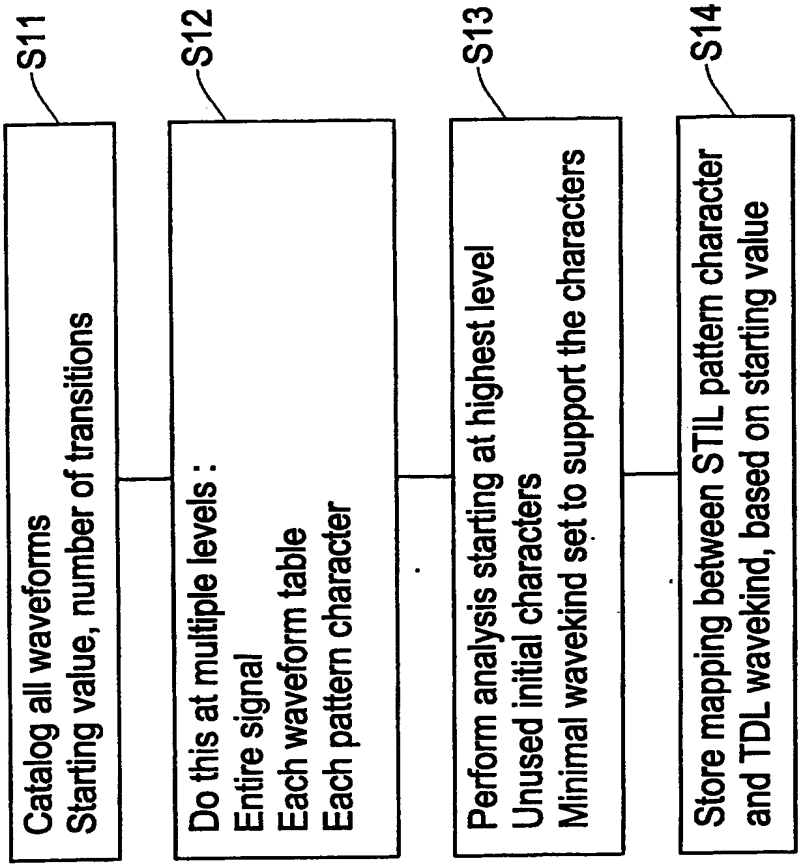

S11
Catalog all waveforms
Starting value, number of transitions
S12
Do this at multiple levels :
Entire signal
Each waveform table
Each pattern character
S13
Perform analysis starting at highest level
Unused initial characters
Minimal wavekind set to support the characters
S14
Store mapping between STIL pattern character
and TDL wavekind, based on starting value

Fig. 12

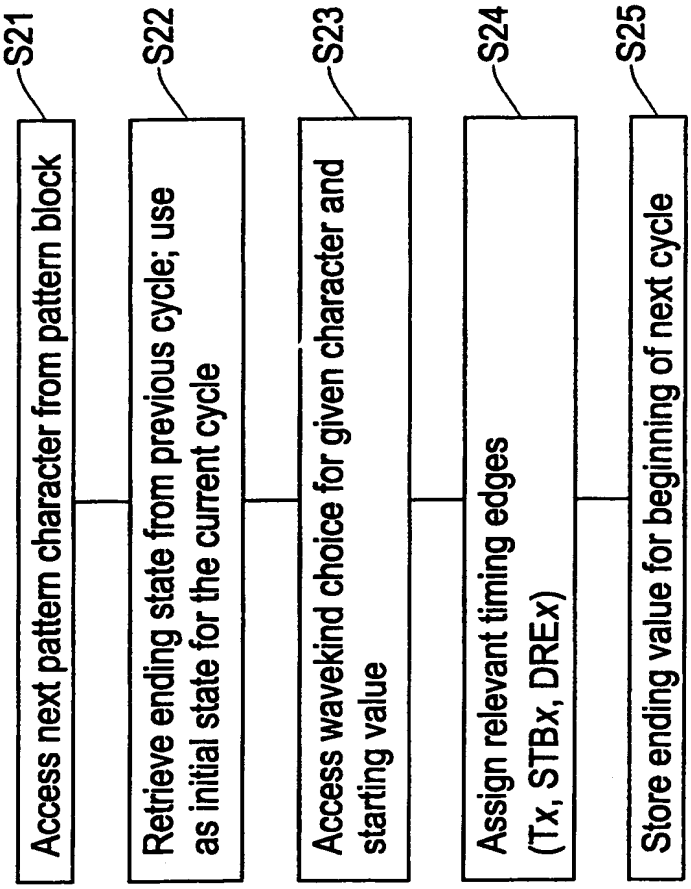
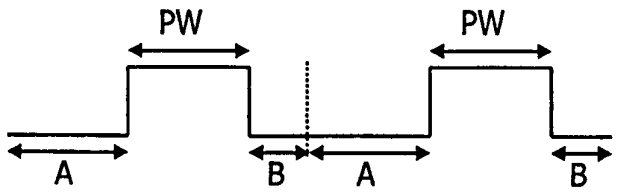
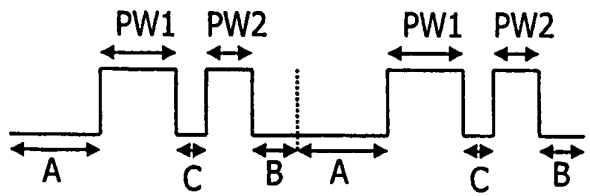

S21
Access next pattern character from pattern block
S22
Retrieve ending state from previous cycle; use as initial state for the current cycle
S23
Access wavekind choice for given character and starting value
S24
Assign relevant timing edges (Tx, STBx, DREx)
S25
Store ending value for beginning of next cycle

Fig. 13
Multi-Clock (MCLK)

- MCLK
  - Find repetitive basic timing unit in STIL pattern char
  - Can be based on single- or double-pulse waveform





  - Determine number of repetitions and factor into Rate

PCT/US00/26189

Fig. 14
Pin Mux